Interface Design for Interoperability for the Land Information System

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN)

CAN-00OES-01

Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

Version 1.1

Draft 2

Contents

List of Figures

1 Introduction

This document describes the design policy for interoperability for the Land Information System (LIS) [5] implemented under funding from NASA's ESTO Computational Technologies Project. This design is submitted to satisfy the Task Agreement GSFC-CT-2 under Cooperative Agreement Notice CAN-00-OES-01 increasing interoperability and performance of grand challenge applications in the earth, space, life, and microgravity sciences.

Code interoperability is important not only between components of a research application, but also between different applications, to decrease the cost of development. Research applications with reusable components facilitate faster development of future applications and enables a broader user base.

This document outlines two different types of interoperability that LIS intends to define and adopt:

• Internal Interoperability: Provide an interoperable framework for the land surface modeling community by defining adaptive, flexible interfaces for incorporating new land surface models into LIS. This will be done by reorganizing the central driver of the LIS, the Global Land Data Assimilation System (GLDAS) software, to take advantage of new object oriented-like strategy of Fortran 90 that will facilitate efficient incorporation of land surface models other than the three models targeted for incorporation in LIS.

• External Interoperability: Participate with Earth, space, life, and microgravity scientific communities by adopting the utilities and compliance guidelines provided by the Earth System Modeling Framework (ESMF) [3]. This will allow LIS to couple and communicate with ocean, atmospheric, and climate communities that can use the outputs of land surface models. LIS will also comply with established land surface modeling standards such as Assistance for Land Modeling Activities (ALMA) [1].

2 Land Surface Modeling in LIS

In general, land surface modeling seeks to predict the terrestrial water, energy, and biogeochemical processes by solving the governing equations of soil-vegetation-snowpack medium. Land surface modeling combined with data assimilation seeks to synthesize data and land surface models to improve our ability to predict and understand these processes. The ability to predict terrestrial water, energy, and biogeochemical processes is critical for applications in weather and climate prediction, agricultural forecasting, water resources management, hazard mitigation and mobility assessment. In order to predict water, energy, and biogeochemical processes using (typically 1-D vertical) partial differential equations, land surface models require three types of inputs: (1) initial conditions, which describe the initial state of land surface; (2) boundary conditions, which describe both the upper (atmospheric) fluxes or states, also known as "forcings" and also the lower(soil) fluxes or states; and (3) parameters, which are a function of soil, vegetation, topography, etc., and are used to solve the governing equations.

LIS uses the GLDAS [4] model control and input/output system that drives multiple offline one-dimensional land surface models (LSMs) to facilitate global land surface modeling within a data assimilation system framework. LIS is expected to include three different land surface models, namely, CLM [2], NOAH [6], and VIC [7]. The GLDAS driver in LIS uses various satellite and ground based observation systems within a land data assimilation framework to produce optimal output fields of land surface states and fluxes. In addition to being forced with real time output from numerical prediction models and satellite radar precipitation measurements, GLDAS derives model parameters from existing topography, vegetation and soil coverage. The model results are aggregated to various temporal and spatial scales, e.g., 3 hourly, 0.25 deg x 0.25 deg.

The execution of GLDAS starts with reading in the user specifications. The user selects the model domain and spatial resolution, the duration and time step of the run, the land surface model, the type of forcing from a list of model and observation based data sources, the number of "tiles" per grid square, the soil parameterization scheme, reading and writing of restart files, output specifications, and the functioning of several other enhancements including elevation correction and data assimilation. The LSMs in GLDAS are driven by atmospheric forcing data such as precipitation, radiation, wind speed, humidity, etc., from various sources. GLDAS applies spatial interpolation to convert forcing data to the appropriate resolution required by the model. Since the forcing data is read in at regular intervals, GLDAS also temporally interpolates time average or instantaneous data to that needed by the model at the current time step. Figure1 shows the structure of GLDAS.

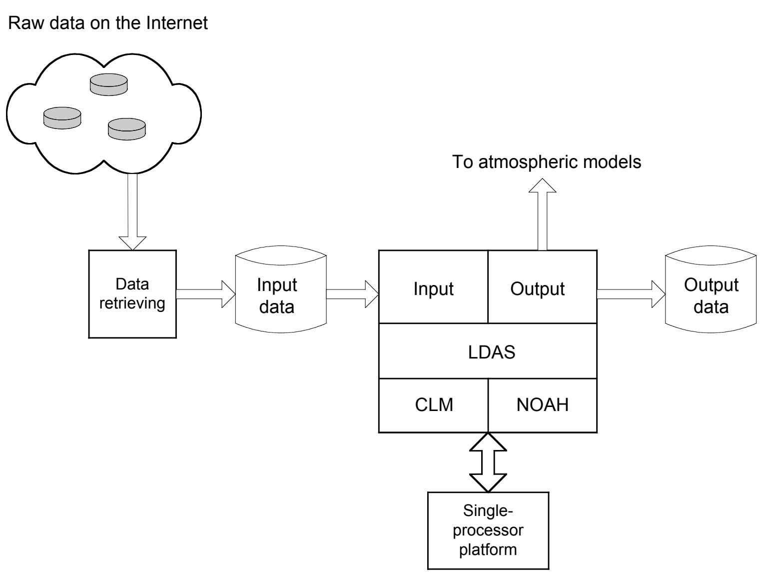

Figure 1: Structure of GLDAS

3 Internal Interoperability in LIS

The GLDAS driver in LIS is redesigned, making use of the advanced features of

Fortran 90 programming language, which are especially useful for object oriented programming. The modified driver is designed using object oriented design principles, providing a number of well-defined interfaces or "hook points" for enabling rapid prototyping and development of new features and applications into LIS.
Figure 2 shows the organization of modules and the main drivers in GLDAS. The main driver that initializes other modules is represented by **ldasdrv**. **ldasdrv** initializes parallelization routines through pool_module and the land surface modeling routines through **ldasdrv_module**. **ldas_module** contains the variables for LSM initializations, executions and outputs. The representation and management of time is encompassed in **time_module** and **grid_module** contains the variables used for spatial grid representation. **baseforcing_module** includes interfaces that are used to incorporate different atmospheric and observation forcings. Similarly, **lsm_module** provides interfaces that can be extended to incorporate new LSMs.

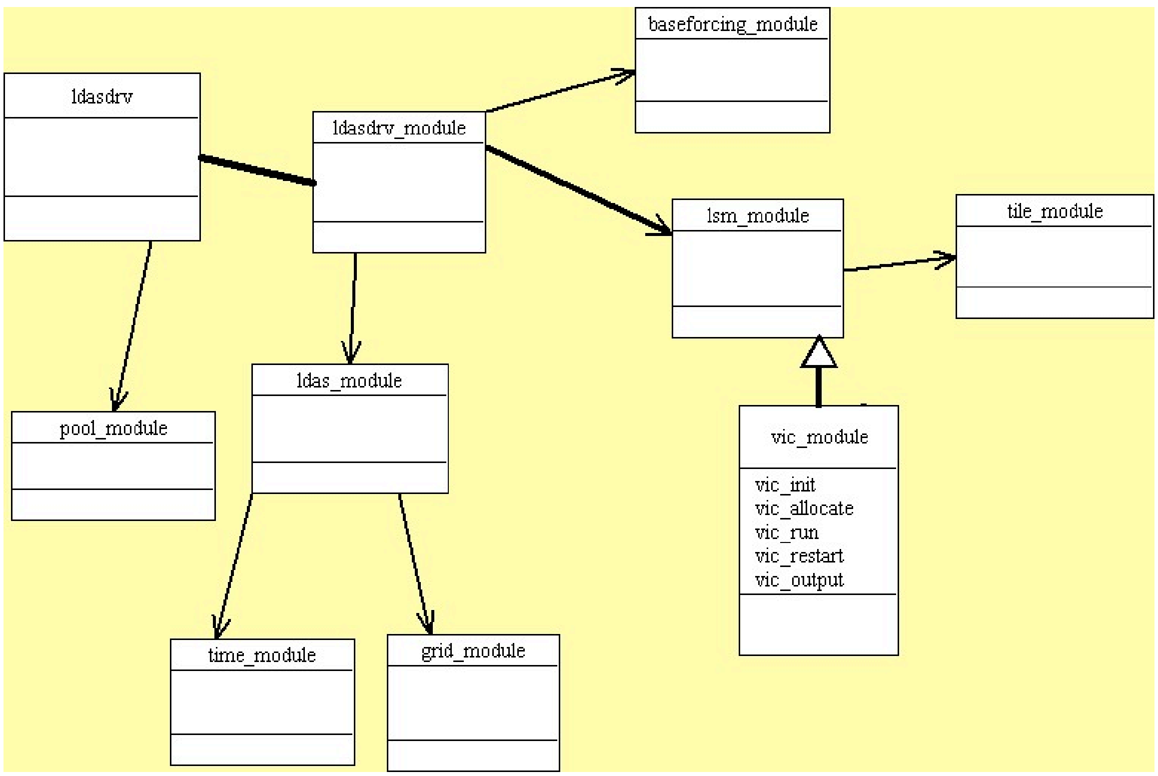

Figure 2: Structure of modules in GLDAS driver

The control flow in GLDAS is shown in Figure 3. In order to define well-defined interfaces that facilitates extensibility for additional features, it is necessary to delegate the flow of control to a number of explicit interfaces and routines. For example, the call to initialize base forcing is delegated to the baseforcing_module from ldasdrv and ldasdrv_module. Similarly, a call to execute an LSM run is transferred to lsm_module from ldasdrv and ldasdrv_module. Figure 4 shows the interfaces and routines defined in baseforcing_module and lsm_module. The get_baseforcing routine in baseforcing_module is used to call the appropriate forcing such as getgeos or getgdas etc., at run time. In a similar fashion, the run-lsm method delegates the LSM execution to run models such as CLM, NOAH, or VIC.

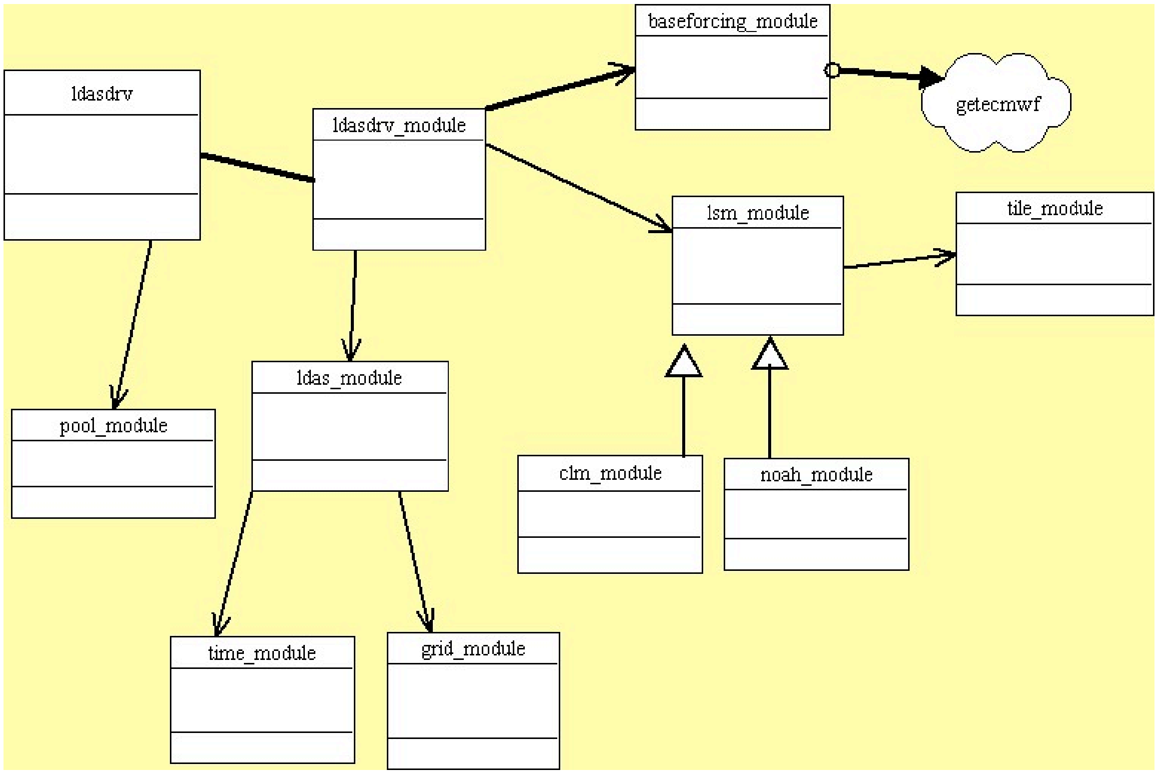

Figure 3: Control Flow in GLDAS

The design of GLDAS driver presented above achieves encapsulation of data and control. The underlying representation does not need to be changed to incorporate a new forcing or a new LSM. The code also simulates polymorphism by allowing the initializations and executions to be determined at runtime. Together, these concepts help to organize the code, making them more flexible, maintainable, and extensible.

baseforcing_module.f
gdas :: private, integer, pointer geos :: private, integer, pointer eta :: private, integer, pointer ncep :: private, integer, pointer nasa :: private, integer, pointer ecmwf :: private, integer, pointer
forcing_ init(this, force) get_ baseforcing(ldas ldas,this,grid)

lsm _module
private, C :: clmdec , pointer clm (:) :: clmdec , pointer private, N :: noahdec, pointer noah(:) :: (:) noahdec, pointer tile(:) :: tiledec , pointer

```
init (this, (LSM )
lsm  _tile_allocate(this, nch)
lsm  _setup( ldas, this)
lsm  _init_output( _ldas, this)
lsm  _green_ alb (ldas, this)
run_ lsm _seq (t, ldas, this)
run_ lsm (pool, nch, ldas ldas, this)
lsm  _readrestart (i, (ldas,grid,this)
write_output( ldas, grid, this)
```

Figure 4: Modules baseforcing-module and lsm-module

4 External Code Interoperability in LIS

To demonstrate interoperability with other scientific modeling communities, LIS will comply with the ALMA data exchange convention and employ the utilities and flexible interfaces provided by ESMF.

4.1 ESMF

The purpose of ESMF is to develop a framework to enhance the ease of use, performance portability, interoperability, and reuse in climate, numerical weather prediction, and data assimilation applications. ESMF is intended to provide a structured collection of building blocks that can be customized to develop model components. ESMF can be broadly viewed as consisting of an infrastructure of utilities and data structures for building model components and a superstructure for coupling and running them.

ESMF provides a utility layer that presents a uniform interface for common system functions. Some of the utilities include time manager, basic communications, error handler, diagnostics, machine model that provides abstractions of hardware and software, etc. LIS intends to use a number of these utilities as and when their implementations are complete.

ESMF also defines a number of guidelines for applications that are intended to be coupled. For gridded components, ESMF is expected to provide standard methods for components to be initialized in parallel configurations and destroyed. LIS will also aim at implementing these interfaces so that LIS can be coupled with other earth system models through ESMF as well as use the utilities provided for gridded components.

LIS currently uses the ESMF time management utility that provides useful functions for time and data calculations, and higher level functions that control model time stepping and alarms. The time management routines in LIS is implemented by using the ESMF time management functions. The time-module in Figure 2 contains functions that initialize, and delegates function calls to use the ESMF time management library functions.

4.2 ALMA Interfaces in LIS

ALMA is a data exchange convention to facilitate the exchange of forcing data for LSM and the results produced by these schemes. The ALMA scheme enables intercomparisons of land surface schemes and ensures that the implementation of procedures to exchange data needs to be done only once. ALMA provides a list of variables needed to force LSMs and a summary of output variable definitions for LSM intercomparisons.

By implementing the ALMA convention in the GLDAS driver, LIS can exchange data with other land surface modeling systems that are also ALMA compliant. Further, it will enable the use of LIS for intercomparison of land surface models for high resolution global modeling.

In order for LIS to be ALMA compliant, a number of interfaces need to be defined as shown in Figure 5. The forcing data is fetched from various locations on the internet, and after preprocessing is fed to the GLDAS driver, which in turns controls the execution of different LSMs. The input interface is expected to convert the forcing data into an ALMA compliant form. The ALMA wrappers for each LSM is expected to perform the translation of GLDAS driver variables to the LSM variables. The output interface is intended to convert the outputs from various LSMs into an ALMA format. Various design issues for these interfaces are discussed below.

4.2.1 Input Interface

Global atmospheric model predictions provide baseline forcing for GLDAS, but whenever possible, the modeled fields are replaced or corrected by observation-based fields. The global data currently in various different data formats. The preprocessing routines for input data will convert the fetched data from internet into a self describing data format such as netcdf/grib. The Input interface will make use of the metadata information present in these data files along with the input forcing ALMA definitions

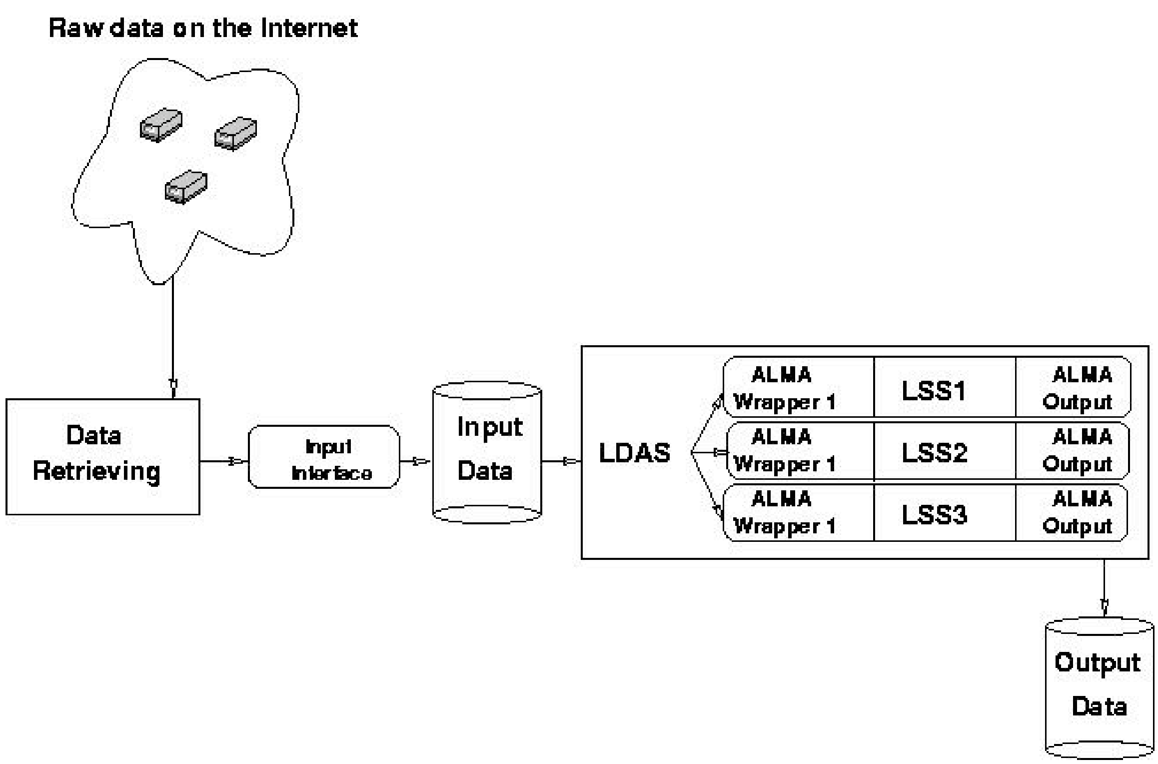

Figure 5: Proposed ALMA interfaces in LIS

to generate an ALMA compliant format. Special attention needs to be paid to the following issues.

• Units : All the required information to convert a forcing variable to the ALMA definition form need to be supplied. An exhaustive list of possibilities for the each forcing field need to be defined. Some special cases might also include dimensionless variables. For e.g, all the required information to convert a relative humidity value to a specific humidity value need to be specified.
• Direction : The sign conventions for each variable definition need to be converted to the ALMA format. Some forcing schemes might define a positive sign to be for an exchange from land to atmosphere, whereas another might consider positive sign to be for a downward direction from the sun to the earth. An exhaustive list of possible fields that need to covert a given directional definition to the ALMA format need to be specified.
• Other issues: The GLDAS driver might require some variables (derived or otherwise) that does not fall within the current definition of input ALMA definition.

Table 1: Mapping of Forcing variables to NOAH input variables

ALMA Variable	Units	Sign (direction of positive values)	NOAH variable	Units	Sign	Required Conversion
Wind-N	M/s	Northward	VWIND	M/s	Northward	
Wind_E	M/s	Eastward	UWIND	M/s	Eastward	
Rainf	Kg/m^2s	Downward	PRCP	Mm/s	Downward	
Snowf	Kg/m^2s	Downward				
Tair	K		SFCTEMP	K		
Qair	Kg/kg		Q2			
Psurf	Pa		SFCPRS	Pa		
SW down	W/m^2	Downward	SOLDN	W/m^2	Downward	
LW down	W/m^2	Downward	LWDS	W/m^2	Downward	
LSRainf	Kg/m^2s					
C Rain f	Kg/m^2s		CPCP	Mm/s		
C SNOW f	Kg/m^2s					
LS Snow f	Kg/m^2s					
SW Rain f	$(Kg/m^2s)^2$					
SV Snow f	$(Kg/m^2s)^2$					
Wind	M/s		SFCSPD	M/s		

The input interface will provide these variables. Compliance to the ALMA standard is considered to be providing all the mandatory variables that are specified in the definition.

4.2.2 ALMA Wrappers

Each LSM scheme included in LIS is expected to be capable of receiving variables in the ALMA form. The ALMA wrappers for each LSM will perform the required conversion from GLDAS driver variables to LSM variables in accordance with the ALMA format. The design includes a list of conversions required. Table 1 and 2 list a mapping of forcing data to NOAH and VIC input variables, respectively.

4.2.3 Output Interface

Defining a generic output interface that converts output variables from different LSMs to an ALMA format is difficult, since explicit information is required to do the mapping from an LSM variable to a corresponding ALMA output variable. One of the intents of the ALMA standard is to put the onus of complying to the ALMA output

Table 2: Mapping of Forcing variables to VIC input variables

ALMA Variable	Units	Sign (direction of positive values)	VIC variable	Units	Sign	Required Conversion
Wind-N	M/s	Northward				
Wind_E	M/s	Eastward				
Rainf	Kg/m^2s	Downward	PREC	Mm/s	Downward	X3600
Snowf	Kg/m^2s	Downward				
Tair	K		AIR_TEMP	C		-273.16
Qair	Kg/kg					
Psurf	Pa		PRESSURE	kPa		X0.001
SW down	W/m^2	Downward	SHORTWAVE	W/m^2	Downward	
LW down	W/m^2	Downward	LONGWAVE	W/m^2	Downward	
LSRainf	Kg/m^2s					
C Rain f	Kg/m^2s					
C SNOW f	Kg/m^2s					
LS Snow f	Kg/m^2s					
SW Rain f	$(Kg/m^2s)^2$					
SV Snow f	$(Kg/m^2s)^2$					
Wind	M/s		WIND	M/s	-	-

Table 3: Mapping of ALMA and VIC output variables: General Energy Balance

ALMA variable	Units	SIGN (direction of positive values)	Priority	VIC variable	Units	Sign	Status
Swnet	W/m^2	Downward	M	Net_short	W/m^2	Downward	Y
Lwnet	W/m^2	Downward	M				
Qle	W/m^2	Upward	M	Latent	W/m^2	Upward	Y
Qh	W/m^2	Upward	M	Sensible	W/m^2	Upward	Y
Qg	W/m^2	Downward	M	Grnd_flux	W/m^2	Downward	Y
Qf	W/m^2	Solid to Liquid	R				I
Qv	W/m^2	Solid to Vapor	O				
Qtau	N/m^2	Downward	R				I
Qa	W/m^2	Downward	O	Advection	W/m^2	Downward	Y
DelSurfHeat	J/m^2	Increase	R				I
DelColdCont	J/m^2	Increase	R	deltaCC	J/m^2	Increase	Y

standard on the LSMs so that intercomparisons between them can be done seamlessly. LIS will adopt this philosophy, assuming that the LSMs are ALMA compliant.

ALMA output standard lists a number of mandatory variables that are required to do water and energy balance. The output interface will use these variables to compute water and energy balance calculations for different LSMs. The output of recommended and optional variables will depend on the LSM employed.
A mapping between a list of ALMA output variables and VIC variables are presented in Tables 3 to 10.

The ALMA standard categorized each ALMA variable into a priority category which appears in Tables 3 to 10 under the heading Priority. The priority indicates whether the variable is a mandatory (M) output to comply with the standard, recommended (R) to comply with the standard, or optional (O) to comply with the standard.

The status category in Tables 3 to 10 indicates the current status of the ALMA variable in the land surface model. A yes (Y) indicates that the ALMA mandatory variable is currently output from the model. A no (N) indicates that the ALMA mandatory variable does not exist is the current output from the model and would require changes in model code to calculate and/or output the variable. A intermediate (I) variable indicates that the variable is not part of the model output currently.

5 Final Remarks

The goal of LIS is to develop a leading edge land surface modeling and data assimilation system to support broad land surface research and application activities, to help define earth system modeling interoperability standards, and to lead the effective application of high performance computing to high-resolution, real-time earth system studies. The framework oriented design of LIS presented in this document and the use and adoption of standards such as ESMF and ALMA helps in providing a platform for land surface modelers and researchers. The flexible, adaptable interfaces in LIS helps to ease the cost of development of new applications. Utilities such as tools for high performance computing helps the researchers in rapid prototyping and development. Further, participation in the standards laid out by ESMF also helps in coupling with other earth system models.

Table 4: Mapping of ALMA and VIC output variables : General Water Balance

ALMA variable	Units	Sign (direction of positive values	Priority	VIC variable	Units	Sign	Status
Snowf	Kg/m^2s	Downward	M				I
Rainf	Kg/m^2s	Downward	M	Prec	Mm/hr	Downward	Y
Evap	Kg/m^2s	Upward	M	Evap	Mm/hr	Upward	Y
Qs	Kg/m^2s	Out of Grid cell	M	Runoff	Mm/hr	Out of grid cell	Y
Qred	Kg/m^2s	Into grid cell	O				N
Qsb	Kg/m^2s	Out of grid cell	M	baseflow	Mm/hr	Out of grid cell	Y
Qsm	Kg/m^2s	Solid to liquid	M				I
Qfz	Kg/m^2s	Liquid to solid	M				I

Qst	Kg/m^2s		R				I
DelSoilMoist	Kg/m^2	Increase	M				
DelSWE	Kg/m^2	Increase	M				
DeslSurFStor	Kg/m^2	Increase	M				I
DelIntercept	Kg/m^2	Increase	R				I

Table 5: Mapping of ALMA and VIC output variables : Surface State Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
SnowT	K	-	M				I
VegT	K	-	M				I
BareSoilT	K	-	M				I
AvgSurfT	K	-	M	surf_temp	C	-	Y
RadT	K	-	M	rad_temp	K	-	Y
Albedo	-	-	M	albedo	-	-	Y
SW E	Kg/m^2	-	M	swq	Mm	-	Y
SW Eveg	Kg/m^2	-	O	snow_canopy	Mm	-	Y
SirfStpr	Kg/m^2	-	M				

Table 6: Mapping of ALMA and VIC output variables : SubSurface State Variables

ALMA variable	Units	Sign (direction of positive values	Priority	VIC variable	Units	Sign	Status
SoilMoist	Kg/m^2	-	M	moist	mm	-	Y
SoilTemp	K	-	R				I
SMLiqFrac		-	O				I
SMFrozFrac		-	O				I
SoilWet		-	M				I

Table 7: Mapping of ALMA and VIC output variables: Evaporation Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
PotEvap	Kg/m^2s	Upward	R				I
Ecanop	Kg/m^2s	Upward	R	Evap_canop	Mm/hr	Upward	Y
Tveg	Kg/m^2s	Upward	M	Evap_veg	Mm/hr	Upward	Y
Esoil	Kg/m^2s	Upward	M	Evap_bare	Mm/hr	Upward	Y
Ewater	Kg/m^2s	Upward	R				I

RootMoist	Kg/m^2		M				N
CanopInt	Kg/m^2		R	Wdew	Mm/hr		Y
EvapSnow	Kg/m^2		R	Sub_snow	Mm/hr		Y
SubSnow	Kg/m^2s		R	Sub_canop	Mm/hr		Y
SubSurf	Kg/m^2s		R				I
ACond	M/s		M				I

Table 8: Mapping of ALMA and VIC output variables : Other hydrologic Variables

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
Dis	M^3/s		O				I
WaterTableD	m	-	O				N

Table 9: Mapping of ALMA and VIC output variables : Cold Season Processes

ALMA variable	Units	Sign (direction of positive values)	Priority	VIC variable	Units	Sign	Status
SnowFrac	-	-	O				I
RainSnowFrac	-	-	O				I
SnowfSnowFrac	-	-	O				I
IceFrac	-	-	O				I
IceT	M	-	O				I
Fdepth	M	-	O				I
Tdepth	M	-	O				I
Salbedo	-	-	R				I
SnowTProf	K	-	R				I
SnowDepth	M	-	R	Snow_depth	cm	-	Y
SlipFrac	-	-	R				I

Table 10: Mapping of ALMA and VIC output variables : Variables to be compared

ALMA variable	Units	Sign (direction of positive values)	Priotity	VIC variable	Units	Sign	Status
LW up	W/m^2	Upward	R				

References

[1] ALMA. http://www.lmd.jussieu.fr/ALMA.

[2] CLM. http://www.cgd.ucar.edu/tss/clm/

[3] ESMF. http://www.esmf.ucar.edu.

[4] GLDAS. http://ldas.gsfc.nasa.gov.

[5] LIS. http://lis.gsfc.nasa.gov.

[6] NOAH. ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noahlsm/.

[7] VIC. http://hydrology.princeton.edu/research/lis/index.html.